Claims

What is claimed is:

1. A method for use in establishing a secure exchange of information between an end user and a server in a distributed network environment, the method comprising the steps of:

in accordance with a network-based controller supervisor associated with the server and in communication with the end user and an application in the server, wherein the controller supervisor supervises processes in response to user requests, the controller supervisor:

invoking a service function in a separate service thread after a successful server connection, wherein the service thread enables user request mapping; and

invoking an initialize function at server initialization for launching at least one persistent thread that facilitates communication between at least one service thread and a corresponding process.

2. The method of claim 1, wherein the at least one persistent thread comprises:

a communication thread that facilitates communication between the at least one service thread and the corresponding process; and

a request thread that facilitates communication for special requests for processes.

3. The method of claim 1, wherein user request mapping comprises the steps of:

maintaining a process table and a session table; and

validating certificates.

4. The method of claim 3, wherein the step of validating certificates comprises the steps of:

mapping a presented certificate to a user ID; and

mapping a request to a validator if a certificate is not presented.

5. The method of claim 4, wherein the step of maintaining a process table and a session table comprises the steps of:

determining if the user ID exists in the session table;

creating a row in the session table if the user ID does not exist and blocking the service thread responding to the user request until the corresponding process has been launched; and

sending a message to a daemon to generate a password for the end user based on the user ID.

6. The method of claim 5, further comprising the steps of:

determining whether the corresponding process is on the process table; and

launching the corresponding process by the daemon if the corresponding process is not on the process table.

7. The method of claim 6, further comprising the steps of:

sending the user request to the launched corresponding process;

launching a process thread linked to the service thread to process the user request; and

returning resulting data to the controller supervisor and the end user.

8. The method of claim 6, further comprising the steps of:

sending a message to the communication thread having a token identifying the service thread;

verifying the token and populating the corresponding process table entry with the token; and

unblocking the service thread which then sends the user request to the corresponding process mapped to the user ID which launches a process thread linked to the service supervisor thread to process the user request.

9. The method of claim 5, further comprising the steps of:

adding the user ID to a password file;

creating a home directory;

changing ownership for a directory and its contents to a new user ID;

launching the process under the new user ID; and

forwarding the password to use to protect the private keys through a pipe.

10. The method of claim 1, further comprising the step of processing the user request without a certificate.

11. Apparatus for use in establishing a secure exchange of information between an end user and a server in a distributed network environment, the apparatus comprising:

a network-based controller supervisor associated with the server and in communication with the end user and an application in the server, wherein the controller supervisor supervises processes in response to user requests, operative to: (i) invoke a service function in a separate service thread after a successful server connection, wherein the service thread enables user request mapping; and (ii) invoke an initialize function at server initialization for launching at least one persistent thread that facilitates communication between at least one service thread and a corresponding process.

12. The apparatus of claim 11, wherein the at least one persistent thread comprises:

a communication thread that facilitates communication between the at least one service thread and the corresponding process; and

a request thread that facilitates communication for special requests for processes.

13. The apparatus of claim 11, wherein user request mapping is operative to:

maintain a process table and a session table; and

validate certificates.

14. The apparatus of claim 13, wherein the operation of validating certificates is operative to:

map a presented certificate to a user ID; and

map a request to a validator if a certificate is not presented.

15. The apparatus of claim 14, wherein the operation of maintaining a process table and a session table is operative to:

determine if the user ID exists in the session table;

create a row in the session table if the user ID does not exist and blocking the service thread responding to the user request until the corresponding process has been launched; and

send a message to a daemon to generate a password for the end user based on the user ID.

16. The apparatus of claim 15, further operative to:

determine whether the corresponding process is on the process table; and

launch the corresponding process by the daemon if the corresponding process is not on the process table.

17. The apparatus of claim 16, further operative to:

send the user request to the launched corresponding process;

launch a process thread linked to the service thread to process the user request; and

return resulting data to the controller supervisor and the end user.

18. The apparatus of claim 16, further operative to:

send a message to the communication thread having a token identifying the service thread;

verify the token and populate the corresponding process table entry with the token; and

unblock the service thread which then sends the user request to the corresponding process mapped to the user ID which launches a process thread linked to the service supervisor thread to process the user request.

19. The apparatus of claim 15, further operative to:

add the user ID to a password file;

create a home directory;

change ownership for a directory and its contents to a new user ID;

launch the process under the new user ID; and

forward the password to use to protect the private keys through a pipe.

20. The apparatus of claim 11, further operative to process the user request without a certificate.

21. An article of manufacture for use in establishing a secure exchange of information between an end user and a server in a distributed network environment,

comprising a machine readable medium containing one or more programs which when executed implement the steps of:

in accordance with a network-based controller supervisor associated with the server and in communication with the end user and an application in the server, wherein the controller supervisor supervises processes in response to user requests, the controller supervisor:

invoking a service function in a separate service thread after a successful server connection, wherein the service thread enables user request mapping; and

invoking an initialize function at server initialization for launching at least one persistent thread that facilitates communication between at least one service thread and a corresponding process.